음성 데이터 보안을 위한 효율적인 ECC 암호 알고리즘 설계 및 구현

김현수* · 박석천**

Design and Implementation of effective ECC Encryption Algorithm for Voice Data

Hyun-Soo Kim* · Seok-Cheon Park**

요 약

모바일 인터넷전화는 요금이 무료일 뿐만 아니라 편리한 인터페이스와 일반전화 급의 통화품질을 제공하고 있어 많은 사용자들이 선호하고 있다. 반면 인터넷망을 이용하기 때문에 제3자에 의해 도청의 위험을 가지고 있어 음성 데이터에 대한 보안이 중요시 되고 있다. 기존에는 음성 데이터 보안에 대칭키 암호 알고리즘이 많이 이용되었지만, 공개키 방식의 ECC 암호 알고리즘은 다른 공개키 방식의 알고리즘보다 비트 대비 암호화 강도가 뛰어나기 때문에 음성 데이터 암호화에 더 선호되고 있다. 그러나 기존 방식은 암호 연산 횟수 때문에 자원 소모가 열악한 모바일 환경에서는 제약이 될 수 있다. 따라서 본 논문에서는 암호 연산 횟수를 감소하여 자원 소모 효율성을 높일 수 있는 방법을 제안한다.

ABSTRACT

Many people is preferred to mVoIP which offers call telephone-quality and convenient UI as well as free of charge. On the other hand, security of mVoIP is becoming an issue as it using Internet network may have danger about wiretapping. Although traditionally encryption algorithm of symmetric key for security of voice data has been used, ECC algorithm of public key type has been preferring for encryption because it is stronger in part the strength of encryption than others. However, the existing way is restricted by lots of operations in poor mobile environment. Thus this paper proposes the efficiency of resource consumption way by reducing cryptographic operations.

키워드

모바일 인터넷전화, 음성 데이터, 타원곡선, 암호 알고리즘

Key word

mVoIP, voice data, ECC, Encryption Algorithm

* 준회원 : 경원대학교 전자계산학과 석사과정 (heavenow@nate.com)
** 종신회원 : 경원대학교 컴퓨터공학 정교수 (교신저자)

접수일자 : 2011. 10. 14
심사완료일자 : 2011. 10. 20

Ⅰ. 서 론

최근 스마트폰을 이용한 인터넷전화가 큰 인기를 끌고 있다. 모바일 인터넷전화(mVoIP:Mobile Voice over Internet Protocol)는 요금이 무료일 뿐만 아니라 일반전화 급의 통화품질을 제공하기 때문이다. 모바일 인터넷전화가 활성화됨에 따라 음성 데이터에 대한 보안이 큰 관심사가 되었다. 인터넷망을 이용하기 때문에 도청의 피해가 존재하기 때문이다. 음성 데이터의 보안을 위해 기존에는 대칭키 방식의 암호 알고리즘이 주로 이용되었지만, 공개키 방식의 암호 알고리즘은 대칭키 방식보다 뛰어난 보안 강도를 가지고 있기 때문에 보다 더 안전하게 음성 데이터를 전달할 수 있다.

일반적으로 160비트 키 사이즈의 ECC(Elliptic Curve Cryptosystem) 알고리즘은 1,024비트 키 사이즈를 가지는 RSA(Rivest Shamir Adlman) 알고리즘과 대등한 보안 강도를 가지고 있어, 키의 비트 대비 암호화 강도가 뛰어난 ECC암호 알고리즘을 이용하여 음성 데이터를 암호화가 선호된다[1]. 그러나 기존 ECC 암호 알고리즘 방법은 암호화 연산 횟수가 많아 자원 소모량이 많기 때문에 모바일 환경에서는 제약이 될 수 있다. 따라서 본 논문에서는 음성 데이터를 처리하는 음성 코덱의 효율적인 연산 구조 및 암호화 방법을 이용하여 암호화 연산 횟수를 감소할 수 있는 암호화 알고리즘을 설계 및 구현하고, 이에 대한 효율성을 평가한다.

Ⅱ. 관련 연구

2.1 모바일 인터넷전화

모바일 인터넷전화는 모바일 환경에서 사용하는 인터넷전화이며, 음성 데이터를 디지털 데이터로 변환하여 일반 전화회선 대신 인터넷망으로 전송하기 때문에 무료로 사용이 가능하다.

스마트폰과 모바일 인터넷전화 어플리케이션의 출시 및 무선 인터넷망의 확산으로 모바일 인터넷전화 이용자가 증가하고 있는 추세이며, 차세대 통신서비스는 궁극적으로 모바일 인터넷전화를 지향할 전망이다. 모바일 인터넷전화는 통신서비스 수요자들의 요구 수준 변화에 부응할 수 있는 수단으로 전송속도 증대, 효율적인 망구조로 다양한 서비스 제공이 가능해질 것이다[2][3].

2.2 ECC 알고리즘

ECC 알고리즘은 1985년 밀러와 코블리츠가 제안한 타원곡선 암호 시스템으로써 이산대수에서 사용하는 유한체의 곱셈군을 타원곡선군으로 대치한 암호체계로써, 다른 암호체계에 비하여 짧은 키 사이즈로 대등한 안전도를 가지는 것이 큰 장점이다. 또한 다양한 타원곡선을 활용할 수 있음에 따라 안전하고 다양한 암호시스템 설계가 가능하다[4][5].

ECC 알고리즘을 이용한 암호 시스템은 난수와 결합한 공개키를 각 단말에 공유하여 공격자가 유추할 수 없는 비밀키로 동기화하고, 암호화하는 순서로 진행된다. 이와 같은 암호 시스템을 구현하기 위해서는 키 분배 알고리즘과 메시지 암호 알고리즘이 구성되어야 하며, ECC 기반의 키 분배 방식은 ECDH(Elliptic Curve Diffie-Hellman)알고리즘이 대표적이다. ECDH 알고리즘은 유한체위의 Diffie-Hellman 알고리즘을 그대로 타원곡선 위에서 변환한다.

EC-ElGamal 알고리즘은 ECDH 알고리즘을 바탕으로 메시지를 암호화하는 방법이며, 현재 ECC 알고리즘 기반의 암호 기법 중 가장 많이 사용하는 암호 알고리즘이다. 메시지 암호화는 비밀키 계산 이후 단말이 메시지와 비밀키를 연산하여 서버 송신 과정과 서버가 비밀키를 이용하여 암호화된 메시지를 연산하는 과정으로 진행된다[6][7].

ECC는 RSA 암호와 더불어, 미국과 국제 표준 기관인 ANSI(American National Standards Institute)와 IEEE (Institute of Electrical and Electronics Engineers) 등에서 표준 암호로 채택되었으며, 특히 WAP(Wireless Application Protocol) 표준으로 채택되어, 스마트폰 등에 의한 이동통신 환경에서 암호 기능을 효율적으로 처리하는 수단으로 최근 각광을 받고 있다. 이러한 ECC의 새로운 수요에 부응하기 위해, 미국은 2009년 6월 새로운 ECC 표준문서인 FIPS PUB 186-3을 발표하여, ECC 활용을 촉구하고 있다. 국내외 보안업체들 역시 최근 스마트폰 등 의 모바일 장치에서 동작되는 ECC 장치를 개발 및 출시하고 있으며, 효율적인 ECC 구현과 관련된 국제특허들을 출원하고 있는 추세이다[8].

2.3 기존 알고리즘의 문제점

ECC 암호 알고리즘은 타원곡선 위의 좌표간의 연산을 통해 암호화를 수행한다. 즉 숫자 간의 연산이기 때문에 암호화 과정에서 문자열, String 등의 데이터는 사용할 수 없는 구조이다. 그림 1은 기존 음성 데이터 암호화 방식에 대한 알고리즘의 일부를 나타낸다.

```
...
byte data[], returnData[]; //인코딩된 음성 데이터, 암호화된 데이터
String xkey, ykey;  //비밀키 x, y 값

for i from 0 to i<(data.length) by +2: //음성 데이터 연산 반복
    returnData ← scolar(data[i], data[i+1], xkey, ykey);
  //각 배열의 순서대로 비밀키(x,y)와 암호화 연산
  ...
return  returnData;  //암호화 데이터 반환
```

그림 1. 기존 ECC 암호화 알고리즘
Fig. 1 Original ECC encryption algorithm

그림 1과 같이 각 배열의 데이터를 각각 비밀키 x, y와 암호화 연산하여 암호화된 데이터를 반환한다. 이에 따라 음성 코덱의 1회 출력하는 데이터의 크기가 byte data[100]라면, 50회의 스칼라 연산, 즉 암호화 연산을 하게 된다. 스콜라 연산은 결코 가벼운 연산이 아니기 때문에 연산 횟수가 증가하게 되면 자원 소모율이 높아져서 모바일 환경에 큰 제약이 될 수 있다.

Ⅲ. 음성 데이터에 대한 효율적인 ECC 암호 알고리즘 설계

3.1 요구사항 분석

ECC 암호 알고리즘은 타원곡선 위의 주어진 점 P와 양정수 k에 대해 점 $Q = kP$를 계산하는 스칼라 곱셈에 의해 암호화를 처리한다[9]. 따라서 ECC 암호 알고리즘은 타원곡선 위에 존재하는 좌표 값들의 계산이기 때문에 암호화 연산에 필요한 음성 데이터와 비밀키 데이터는 모두 숫자이거나 숫자로의 변환이 필요하다. 또한 제안한 알고리즘의 연산 및 변환을 간소화하기 위해 음수는 양수로, 실수보다는 자연수가 선호된다.

3.2 제안하는 알고리즘 설계

음성 데이터에 대한 암호화 연산의 횟수를 줄이기 위해 바이트 배열의 구조 변경이 필요하다. 음성 데이터가 인코딩되어 바이트 배열에 입력되기 전에 8진수의 데이터로 변경하고, 8진수로 된 데이터를 8구분자를 이용하여 하나의 데이터로 결합한다. 8진수는 8이 사용되지 않기 때문에 8을 이용하여 데이터를 수의 상태로 유지한다. 그림 2는 이에 대한 과정을 나타낸다.

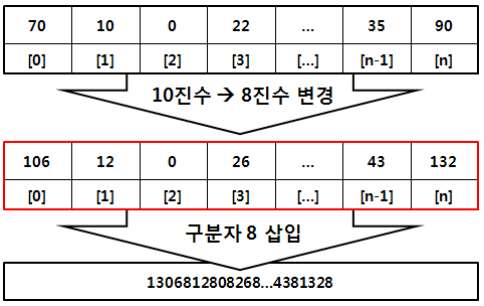

그림 2. 음성 데이터 구조 변경
Fig. 2 Change of voice data structure

구분자를 통해서 구조화된 데이터는 반으로 분할되어 비밀키 좌표 x, y와 암호 연산을 통해 암호화 데이터를 생성한다. 그림 3은 데이터와 비밀키를 통해 암호화 연산하는 과정을 나타낸다.

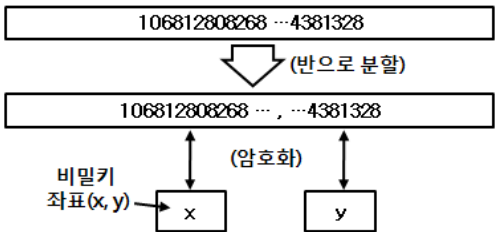

그림 3. 암호화 연산 과정
Fig. 3 Process of cryptographic operations

그림 3과 같은 과정을 통해 음성 코덱에서 음성 데이터를 출력할 때 단 한 번의 암호화 연산과정으로 기존의 알고리즘에 비해 암호화 연산 횟수가 큰 차이로 줄어들었다. 그림 4는 음성 데이터에 대한 암호화 연산 횟수를 감소시킨 알고리즘을 나타낸다.

```
//인코딩된 음성 데이터를 8진수로 변환
            ...
byte data[], returnData[];   //인코딩 data, 암호화된 data
String xkey, ykey;      //비밀키 x, y 값
String xData, yData;  //8진수 입력 값

   for  i  from  0  to  i<(data.length)  by +1:
        if  i<(data.length/2)      //배열 중 앞의 반은 x값
          then  xData ← xData + 8
        else                       //배열 중 나머지 반은 y값
          then   yData ← yData + 8

   returnData ← scolar(xData, yData, xkey, ykey);   //암호연산
```

그림 4. 음성 데이터에 대한 효율적인 ECC 알고리즘
Fig. 4 Efficient ECC algorithms for voice data

음성 데이터를 암호화하기 위해 먼저 코덱으로부터 8진수로 변환된 데이터를 받아 온다. 받아온 데이터를 문자열로 변경하고, 8구분자를 삽입하여 배열 길이의 반은 x값, 나머지 반은 y값으로 나눈다. 구조화된 데이터는 비밀키와 스칼라 연산을 통해 암호화 데이터가 생성된다. 제안한 알고리즘 방법으로 암호화 연산 횟수를 대폭 줄임에 따라 암호화 연산을 위한 자원 소모가 크게 줄기 때문에 모바일 등의 제약적인 환경에서도 효율적인 음성 데이터 암호화 연산이 가능하다.

Ⅳ. 음성 데이터 보안을 위한 효율적인 ECC 암호 알고리즘 구현

4.1 제안 알고리즘 구현

본 논문에서 제안하는 알고리즘은 ECC표준에서 정하는 160비트 이상의 파라미터 값을 가지도록 설정하였다[9]. 따라서 프로그램 구현 시 사전에 정의되어 사용되는 파라미터 값은 표 1과 같다.

표 1. 알고리즘 파라미터 값(10진수)
Table. 1 Algorithm parameter values

구분	값
P	1461501637330902918203684832716283019653785059327
a	1461501637330902918203684832716283019653785059324
b	4
Gx	0
Gy	2

파라미터 P는 타원곡선 좌표의 범위를 나타내는 소수(prime number)로써 P의 크기가 클수록 암호의 강도는 강해진다. 또한 파라미터 a, b는 타원곡선의 성질을 나타낸다. 파라미터 Gx와 Gy는 G라고 하는 좌표의 x와 y값을 나타내며, 같은 비밀키를 생성하기 위해 상대방과 정해놓은 약속된 좌표이다. 위의 파라미터만으로는 제 3자가 암호를 쉽게 풀 수 없기 때문에 공개되어도 무관하다. 본 논문에서 제안하는 알고리즘 구현 환경은 인텔 Duo Core 2.66GHz와 Windows XP Professional 운영체제에서 JAVA JDK 1.6을 사용하였다.

음성 데이터 암호화에 앞서 공개키를 이용한 비밀키 생성이 필요하다. 표 3에서 공개한 P, a, b, 좌표G와 난수를 통해 비밀키를 생성한다. 그림 5는 표 1의 파라미터 값을 바탕으로 비밀키를 구현한 과정을 나타낸다.

```
난수생성(c): 8322
P: 1461501637330902918203684832716283019653785059327
a: 1461501637330902918203684832716283019653785059324
b: 4
좌표G생성 -> (x: 0, y: 2)
** 공개키 생성 **
공개키(x): 1410193119070634967338327905066153822069320652
공개키(y): 938112296925502592071208201091136727319074179 2

** 상대방의 생성 키 **
상대방 난수(k): 8412
공개키(x): 905023977883086172089900879469998587709364169747
공개키(y): 168026939197896500257883247973966855186907581564

** 비밀키 생성 **
[A]비밀키 생성(x): 1146598942075379357917475953154578802191690373 47
[A]비밀키 생성(y): 381188520856845272195843519249057567761987248887
```

그림 5. ECC 암호화를 위한 공개키 및 비밀키 생성
Fig. 5 Generate a public key and private key for ECC encryption

그림 5와 같이 자신(A)의 공개키와 상대방의 공개키를 교환함으로써 비밀키를 생성하였다. 이 비밀키는 음성 데이터와 암호화 연산을 할 때 필요한 키이다. 음성 데이터를 암호화하기 위해 음성 코덱으로부터 인코딩된 데이터를 전달받고, 이를 8구분자를 통해 제안한 알고리즘으로 구조화 하였으며, 결과는 그림 6과 같다.

```
[0]:39 [1]:15 [2]:95 [3]:12 [4]:30 [5]:11 [6]:82 [7]:14 [8]:4 [9]:72 [10]:38
[11]:33 [12]:1 [13]:44 [14]:63 [15]:8 [16]:55 [17]:89 [18]:27 [19]:38 [20]:40
[21]:65 [22]:43 [23]:63 [24]:86 [25]:45

*structed Original Data(x): 8478178137814836813812281684811084684181
*structed Original Data(y): 8548778108678131833846850810185387781268 55
```

그림 6. 음성 데이터의 암호화를 위한 구조화
Fig. 6 Structure for Encryption of voice data

그림 6에서 구조화된 데이터를 암호화 연산을 위해 좌표 구조로 변경하여 구조화된 데이터와 비밀키를 통해 암호화 연산을 수행하면 암호화된 데이터가 생성된다. 이 과정은 그림 7과 같으며, 복호화된 데이터를 살펴보면 구조화된 데이터와 일치함을 알 수 있다.

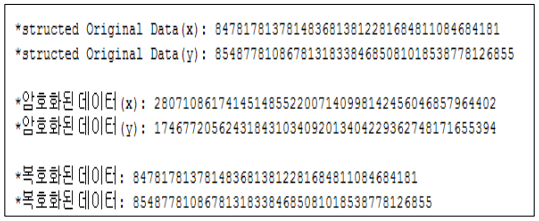

그림 7. 음성 데이터의 암호화와 복호화 연산
Fig. 7 Operation of encryption and decryption of voice data

4.2 제안 알고리즘 평가

본 논문에서 제안하는 알고리즘은 암호화 연산 횟수를 줄임으로써, 연산 시간과 자원 소모율을 줄이고자 하였다. 일반적으로 휴대폰을 이용한 전화의 평균 통화 시간은 1분에서 2분 사이임을 참고하여 제안 알고리즘의 평가는 기존의 음성 데이터에 대한 ECC 암호 알고리즘과 시간별 연산 횟수를 통해 비교하였다. 기존 알고리즘의 암호화 연산의 시간 대비 연산 횟수에 대한 결과는 표 2와 같다.

표 2. 기존 알고리즘의 암호화 연산의 시간 대비 연산 횟수 결과(단위: s\10진수)
Table. 2 Result of the number of original cryptographic operations over time(Decimal)

초\Test	Test1	Test2	Test3	Test4	AVG
30	1143272	1120382	1166319	1145907	1143970
60	2286544	2240764	2332638	2291814	2287940
90	3429816	3361146	3498957	3437721	3431910
120	4573088	4481528	4665276	4583628	4575880
150	5716360	5601910	5831595	5729535	5719850

표 2는 30초 단위로 각각 연산 횟수를 체크한 결과이다. 테스트를 시도할 때마다 여러 변수에 의해 약간의 오차가 있을 수 있기 때문에 몇 번에 걸쳐 테스트를 한 후 평균값을 측정하였다. 표 3은 제안한 알고리즘의 연산 횟수 결과이다.

표 3. 제안한 알고리즘의 암호화 연산의 시간 대비 연산 횟수 결과(단위: s\10진수)
Table. 3 Result of the number of proposed cryptographic operations over time(Decimal)

초\Test	P1	P2	P3	P4	AVG
30	88846	87998	87944	88177	88241
60	177692	175996	175888	176354	176483
90	266538	263994	263832	264531	264724
120	355384	351992	351776	352708	352965
150	444230	439990	439720	440885	441206

표 3과 같이 시간 대비 연산 횟수는 시간이 지날수록 큰 차이를 보이며 격차가 벌어지는 것을 확인할 수 있다.

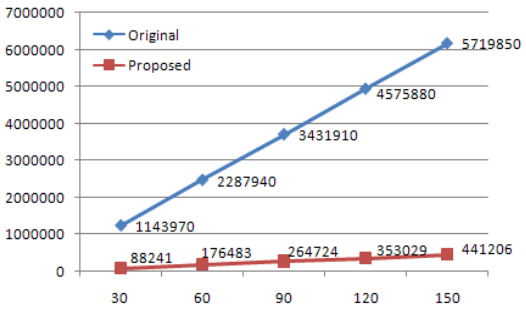

그림 8. 제안한 알고리즘의 연산 횟수 그래프
Fig. 8 Graph of comparison original algorithm with proposed algorithm for the number of operations

그림 8은 기존 알고리즘과 제안한 알고리즘의 연산 횟수를 비교한 그래프이다. 통화 초기 30초에는 기존의 알고리즘과 비교하여 연산 횟수가 약 12배 정도를 유지하지만, 시간이 지날수록 격차가 매우 크게 벌어지는 것을 알 수 있다. 결과적으로 통화 시간이 늘어날수록 연산 횟수가 증가하여 배터리의 소모, CPU(Central Processing Unit)의 사용률 등이 증가하게 됨으로써, 통신 환경의 제약이 있는 스마트폰에 영향을 줄 수 있다. 따라서 제안한 알고리즘은 연산 횟수를 줄임으로써 자원소모율을 감소시켜 보다 효율적인 암호화가 가능하다.

제안한 알고리즘의 다른 특징은 암호화 강도가 높다는 점이다. 기존의 알고리즘을 이용하면 바이트 배열에 저장되어 있는 데이터를 각각 하나씩 암호화해야 한다. 바이트 배열의 데이터는 최소 1비트에서 최대 7비트까지이기 때문에, 타원곡선 좌표의 범위인 P를 ECC에서 지정한 표준 크기인 160비트를 사용하는 것은 매우 비효율적이다. 반면 암호 연산을 위해 P의 크기를 줄인다면 보안의 강도가 낮아질 수 있어 보안의 강도를 높이고자 했던 근본적인 의도에서 벗어날 수 있다. 따라서 제안한 알고리즘을 이용하여 암호 연산을 수행하게 되면 P의 크기는 160비트를 유지하며, 비밀키와 연산하기 위한 구조화된 음성 데이터는 큰 크기를 유지함에 따라 강력한 보안의 강도를 유지할 수 있다.

V. 결 론

스마트폰을 이용한 모바일 인터넷전화가 크게 활성화됨에 따라 모바일 인터넷전화 어플리케이션이 사용자들로부터 인기를 끌고 있다. 스마트폰을 이용한 모바일 인터넷전화는 무료 또는 저렴한 요금과 일반전화 급의 통화품질을 제공하여 유용하게 사용될 수 있다. 이러한 추세와 더불어 인터넷전화에 대한 보안이 크게 이슈가 되고 있다.

기존 인터넷전화의 암호 알고리즘은 주로 대칭키 방식의 암호 알고리즘을 사용하였으나 본 논문에서는 음성 데이터에 대한 보안을 강화하기 위해 공개키 방식인 ECC 암호 알고리즘을 이용하였다. G.계열의 음성 코덱은 바이트 배열 구조에 음성을 처리하기 때문에 배열의 각 데이터를 암호화하게 되면 암호화 횟수에 따른 자원 소모가 많아서 모바일 등의 환경에서는 매우 제약적이다. 따라서 본 논문에서는 배열의 데이터 구조를 변경하여 암호화 횟수를 대폭 줄임으로써 처리 시간과 자원 소모를 대폭 감소시키고자 하였다.

본 논문에서 제안하는 알고리즘을 설계 및 구현하여 기존의 암호 알고리즘과 비교를 통해 암호 연산의 횟수가 약 88% 이상 감소하였음을 확인하였고, 시간이 경과할수록 효과가 크게 나타남을 확인하였다. 또한 기존의 알고리즘과 비교하였을 때 암호화 강도가 높고, 보다 더 효율적임을 알 수 있었다. 따라서 본 논문에서 강조하고자 하였던 암호화 횟수에 따른 자원 소모율과 보안의 강도는 제안한 알고리즘과 비교를 통하여 더 효율적임을 입증하였다. 본 논문에서 구현한 ECC를 이용한 음성 데이터 암호화 알고리즘은 향후 여러 분야에 응용 또는 적용될 것으로 기대된다.

감사의 글

본 연구는 2010년도 산학협동재단의 지원(산학 2010-33 2010.6.30)에 의하여 이루어진 연구입니다. 관계부처에 감사드립니다.

참고문헌

[1] W.Diffie and M.Hellman, ″New directions In Cryptography,″ IEEE Transaction on Information Theory, pp 664~654, 1976.

[2] 박신정, “m-VoIP 전화 현황과 발전 전망,” 전자정보센터, 2009.

[3] 백재영, “Mobile VoIP 시장 전망 및 제품 개발동향,” 전자정보센터, 2009.

[4] 고훈, “타원곡선 알고리즘을 이용한 XML 문서 암호 구현,” 한국인터넷정보학회, 제8권 제1호, 2007. 2.

[5] 김정식, 강부중, 노인우, 임을규, “타원곡선암호 및 구현 방법에 대한 연구,” 한국정보과학회, Vol. 34, No.2(D), 2007.

[6] http://blog.naver.com/santalsm/110035193901

[7] SEC1, ″Elliptic Curve Cryptography,″ Vol. 1.0, pp 62, 2000.

[8] 고승철, 남길현, “타원곡선 암호 구현 WIPO 특허 동향,” 정보보호학회지, 제21권 제5호, 2011. 8.

[9] 김호원, 이석준, 오경희, “센서네트워크 보안 기술 개발 동향,” 정보보호학회지, 제18권 제2호, 2008.

저자소개

김현수(Hyun-Soo Kim)

2010년 경원대 학사
2010년 ~ 현재 경원대
전자계산학과 석사과정

※관심분야 : 네트워크 보안, 모바일 어플리케이션 개발, 암호화 플랫폼 개발, 차세대 인터넷

박석천(Seok-Cheon Park)

1977년 고려대 전자공학과 학사
1982년 고려대 컴퓨터공학 석사
1989년 고려대 컴퓨터공학 박사
1979년 ~ 1985년 금성통신연구소
1991년 ~ 1992년 UC, Irvine Post Doc.
1988년 ~ 현재 경원대학교 컴퓨터공학과 정교수
※관심분야 : 차세대 인터넷, 멀티미디어통신, 네트워크 시큐리티, 모바일 통신